

PENGENDALI POSISI MOTOR DC DENGAN PID MENGGUNAKAN METODE ROOT LOCUS

Oleh :

Agus Nuwolo⁽¹⁾, Adhi Kusmantoro⁽²⁾

agusnuwolo150461@gmail.com, adhiteknik@gmail.com

Fakultas Teknik / Teknik Elektro Universitas PGRI Semarang

Abstrack

Position control using a motor is important at this time, especially in the industry. This control is used to move the conveyor so that it will stop at a designated place. Another application is to move the robot arm to a certain position. The development of control technology is so rapid now allow control is done digitally. The resulting simulation can be used to design a PID controller algorithm. Root locus design results in a PID controller generates 13.6% overshoot and settling time of 0.0386 seconds. With the interference of external PID controller is still able to cope with disturbances. To improve the quality of DC motor position controller is recommended to use logic controller Fuzi (fuzzy logic).

Keywords: control position, Motor DC, Root Locus

I. PENDAHULUAN

Pada sistem pengendali kecepatan permasalahan yang sering terjadi adalah menentukan berapa besar energi listrik yang harus diberikan pada motor supaya berputar pada kecepatan yang diinginkan, bagaimanapun kondisi beban yang digerakan. Pada sistem pengendali posisimasalah utamanya pada berapa lama energi listrik harus diberikan agar motor menggerakkan beban sampai posisi yang diinginkan. Tujuan pengendalian ini adalah agar motor berputar dengan kecepatan yang sesuai dengan yang diinginkan dengan berbagai kondisi beban yang digerakan motor. Terdapat segolongan alat atau sistem lain yang juga

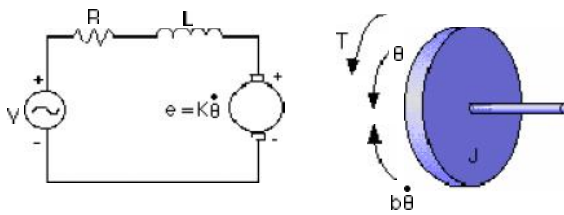
menggunakan motor listrik sebagai penggerak tetapi dengan penggunaan yang berbeda. Pada sistem ini motor digunakan untuk menggerakkan benda kesuatu posisi yang diinginkan. Inilah yang dikenal dengan sistem pengendali posisi. Contoh sistem ini adalah sistem kemudi kapal laut atau pesawat terbang. Pada kebanyakan sistem pengendali kecepatan, yang diinginkan adalah menjaga konstan kecepatan putar untuk segala kondisi beban, tidak mengatur agar kecepatan putarnya berubah-ubah setiap waktu mengikuti masukan acuan yang berubah. Tidak demikian halnya dengan sistem pengendali posisi. Pada sistem ini, akurasi sistem biasanya diukur tidak hanya

dengan *steady-state error* untuk masukan step, tetapi juga dengan *steady-state error* untuk masukan yang berubah dengan waktu. Hal ini biasanya dikenal sebagai *following error*.

II. TINJAUAN PUSTAKA

A. Motor DC

Motor DC bekerja berdasarkan prinsip gaya elektromagnetik sehingga apabila motor tersebut diberi catu daya, arus akan mengalir ke dalam motor kemudian menghasilkan torsi putar yang sebanding dengan arus tersebut. Pemodelan Rangkaian internal Motor DC digambarkan sebagai berikut :



Gambar 2.1 Model motor DC

Persamaan torsi yang dibangkitkan oleh Motor DC dengan pendekatan secara linear menurut persamaan sebagai berikut (Ogata, 1997):

$$T = K \dots \dots \dots (1)$$

dimana K_a adalah konstanta yang karmotorya ngbergantung padabanyaknya lilitan pada ja ngkar, jumlah kutub medan, tipe belitan dan p enampang jangkarnya. Adapun besarnya teg angan ggl induksi la yang dibangkitkan m otorketika berputar adalah sebanding dengan

konstanta motor K_b dan turunan pertama dari posisinya sudut motor

$$(\theta)e = K_b \dot{\theta} \dots \dots \dots (2)$$

Dengan menggunakan hukum newton, bahwa persamaan torsi yang terkait dengan momen inersia dan rasio redamansi motor adalah:

$$T = J\ddot{\theta} + b\dot{\theta} \dots \dots \dots (3)$$

dari persamaan (1) dan (3) diperoleh:

$$i = (J\ddot{\theta} + b\dot{\theta})/K \dots \dots \dots (4)$$

Sedangkan besarnya tegangan V menurut hukum kirchoff adalah:

$$V = iR + L \frac{di}{dt} + e$$

$$V = iR + L \frac{di}{dt} + K_b \dot{\theta} \dots \dots \dots (5)$$

Dengan transformasi laplace persamaan (4) dan (5) maka diperoleh fungsi alih antara posisi sudut motor θ terhadap tegangan armature V dimana $K_a = K_b$

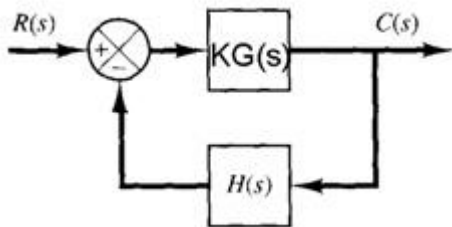
$$\frac{\theta(s)}{V(s)} = \frac{K}{s(Js + b)(Ls + R) + K^2} \dots \dots \dots (6)$$

Persamaan di atas memiliki 5 konstanta yang belum diketahui. Adapun konstanta tersebut adalah sebagai berikut:

1. Momen inersia rotor (J)
2. damping ratio sistem mekanik (b)
3. Konstanta gaya gerak ($K = K_b = K_a$)
4. Hambatan (R)
5. Induktansi (L)

B. Root Locus

Karakteristik tanggapan transient sistem loop tertutup dapat ditentukan dari lokasi pole-pole (Ogata, 1994).



Gambar 2.2 Sistem pengendali kalang tertutup

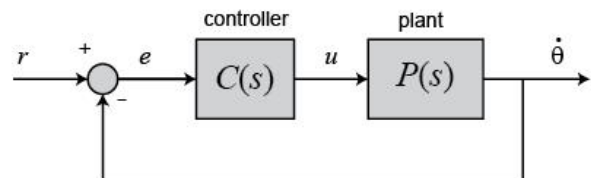
Bila K berubah, maka letak pole-pole nya juga berubah. Untuk mengetahui kawasan letak pole-pole persamaan karakteristik sistem tersebut terhadap kemungkinan kombinasi nilai K , maka diperlukan sebuah metode yang disebut dengan Root Locus. Root berarti akar dan Locus berarti tempat kedudukan. Maka Root Locus adalah tempat kedudukan akar-akar persamaan karakteristik dari sebuah sistem pengendalian proses dengan $K = 0$ sampai $K = \infty$. Ini dapat digunakan untuk menentukan stabilitas sistem tersebut selalu stabil atau ada batas kestabilannya, sehingga dalam merancang sistem kendali, kita bisa mendapatkan hasil *step responses* sesuai yang diinginkan dengan mengubah nilai-nilai K .

III. Metode Penelitian

Bentuk penelitian ini hanya berupa simulasi berbasis komputer dengan alat

bantu *Matlab for window* versi 7.8 dan toolbox jenis kendali. Untuk melakukan perancangan suatu sistem pengendali posisi motor DC melalui tahapan sebagai berikut:

1. Menurunkan persamaan model.

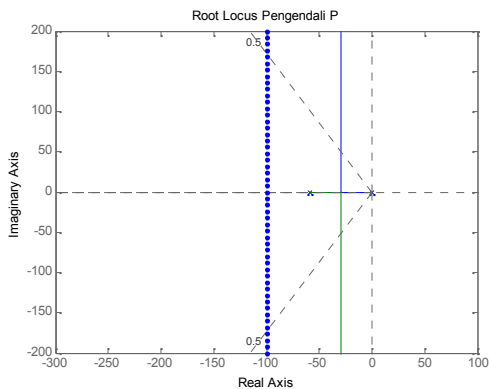


Gambar 3.1 Blok diagram kendali posisi

1. Mendefinisikan karakteristik/ variabel model.
2. Variabel dalam program simulasi adalah sebagai berikut :
 $J = 3.2284 \times 10^{-6} \text{ K} \cdot \text{gm}^2$
 $b = 3.5077 \times 10^{-6} \text{ Nms}$
 $K = 0.0274 \text{ Rad/detik}$
 $R = 4 \text{ Ohm}$
 $L = 2.75 \times 10^{-6} \text{ H}$
3. Melakukan perancangan root locus.
Dalam melakukan perancangan pengendali dengan kriteria overshoot di bawah 16 %, settling time kurang dari 0.04 detik.
4. Melakukan simulasi menentukan root locus untuk pengendali Proporsional, Integral, Proporsional Integral, dan Proporsional Integral Derivative (PID).

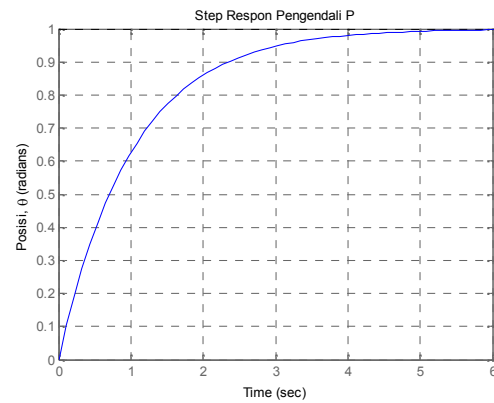
IV. Hasil dan Pembahasan

A. Pengendali Proporsional (P)



Gambar 4.1 Root locus pengendali Proporsional

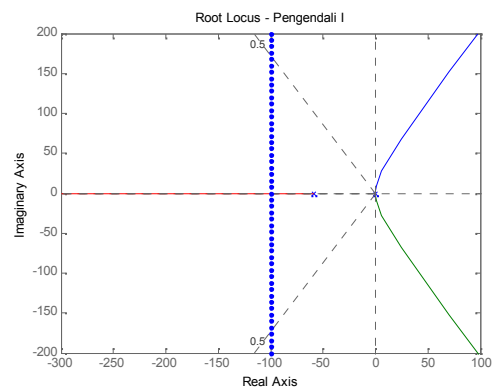
Root locus pengendali proporsional memperlihatkan letak akar dari fungsi alih berada di sebelah kiri sumbu imajiner. Dalam plot diatas garis titik-titik sebesar 60 derajat menunjukkan lokasi pole dengan damping ratio ($\zeta = 0.5$) dan frekuensi alami ($\omega_n = 0$). Antara garis tersebut pole akan punya zeta di atas 0.5 dan di luar garis tersebut mempunyai zeta di bawah 0.5. Dari gambar 4.1 terlihat pengendali ini tidak memiliki zero. Grafik respon pengendali proporsional sulit untuk mencapai nilai set point dengan amplitudo maksimal 0.997 dan settling time 3.99 detik, sehingga stabilitas menjadi masalah untuk jenis pengendali ini.



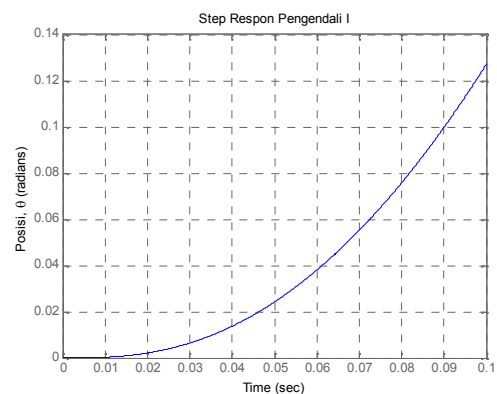
Gambar 4.2 Respon pengendali P

B. Pengendali Integral (I)

Pengendali Integral memperlihatkan respon tidak stabil, pengendali tidak dapat mencapai set point yang terlihat pada gambar 4.4.

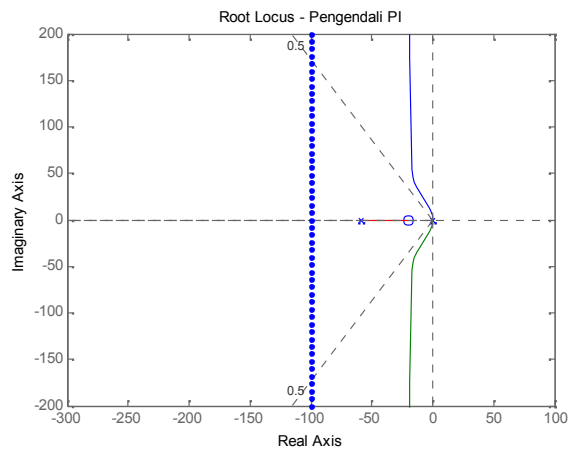


Gambar 4.3 Root locus pengendali I



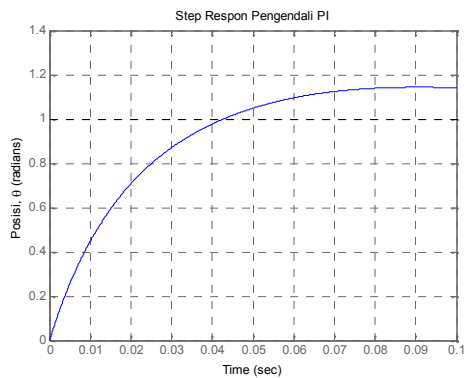
Gambar 4.4 Respon pengendali I

C. Pengendali PI



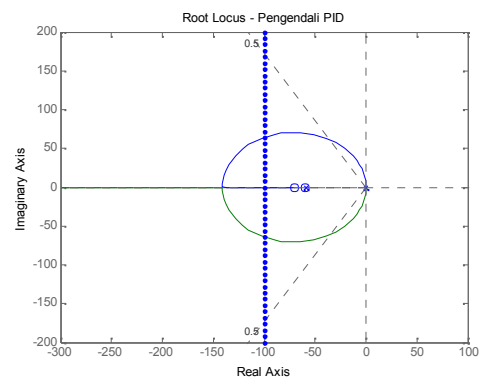
Gambar 4.5 Root locus pengendali PI

Pengendali PI mempunyai zero $s = -20$ dan pole yang terletak di sebelah kiri sumbu imajiner. Respon pengendali memperlihatkan dapat mencapai set point tetapi membutuhkan waktu yang lama. Pengendali ini mempunyai overshoot 14.4 %.



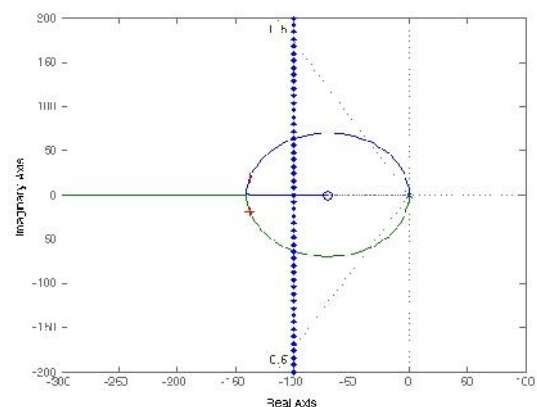
Gambar 4.6 Respon pengendali PI

D. Pengendali PID

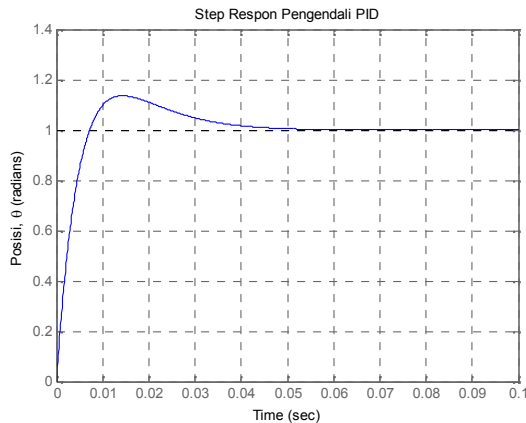


Gambar 4.7 Root locus pengendali PID

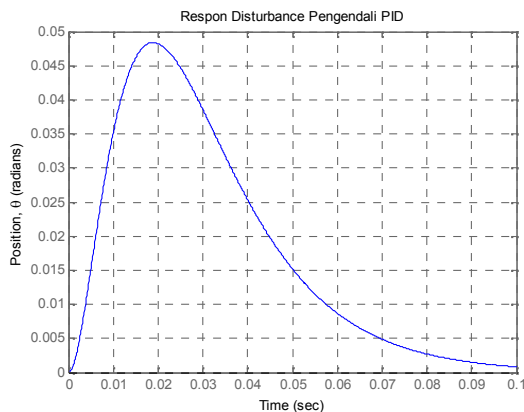
Gambar 4.7 memperlihatkan tempat kedudukan akar pengendali PID yang mempunyai pole $s = -59$ dan zero $s = -60$. Untuk mendapatkan overshoot kurang dari 16 % pole harus berada di antara dua garis titik-titik. Dari plot terlihat pole berada pada lingkaran yang merupakan pole pada posisi yang dapat diterima. Respon sistem pengendali menghasilkan overshoot 13.6 % dan settling time 0.0386 detik, sehingga pengendali PID ini sesuai dengan kriteria perancangan.



Gambar 4.8 Root Locus pengendali PID dengan rlocfind



Gambar 4.8 Respon pengendali PID

Gambar 4.9 Respon pengendali PID
dengan disturbance

Dari grafik respon terlihat overshoot yang terjadi sekitar 14 % dan settling time dibawah 0.04 detik serta tidak terjadi kesalahan steady state. Hasil rancangan sudah sesuai dan gain pengendali yang dihasilkan 0.1308.

V. KESIMPULAN

1. Pole dari root locus bisa lebih dari satu cabang. Pole yang kita tentukan akan berpengaruh terhadap respon sistem.
2. Untuk perancangan root locus pada pengendali Proporsional, Integral dan

Proporsional Integral memperlihatkan respon yang tidak stabil, dengan overshoot dan settling time yang tidak sesuai rancangan.

3. Respon pengendali PID menghasilkan overshoot 13.6 % dan settling time 0.0386 detik. Hasil rancangan root locus sudah sesuai kriteria rancangan.
4. Dengan adanya disturbance respon pengendali mampu mengatasi dengan overshoot dan settling time dibawah nilai yang sudah ditentukan.

VI. DAFTAR PUSTAKA

1. Johnson. Michael, Mohammad H.Moradi,2005,"PID Control : New Identification And Design Method, Springer.
2. Gunterus, Frans: Falsafah Dasar: Sistem Pengendalian Proses, jakarta: PT. Elex MediaKomputindo, Jakarta, 1994
3. Johnson, Curtis: Process Control Instrumentation Technology, Englewood Cliffs, NewJersey, 1988
4. Messner, William and Dawn Tilbury. Control Tutorials for MatLab and Simulink. A Web Based Approach. Addisson Wesley, Inc. 1999.
5. Ogata, Katsuhiko. Modern Control Engineering. 3rd ed. Prentice Hall International. 1997

6. Ogata, Katsuhiko. Solving Control Engineering Problems with MatLab. Englewood Cliffs, New Jersey : Prentice Hall Inc. 1994